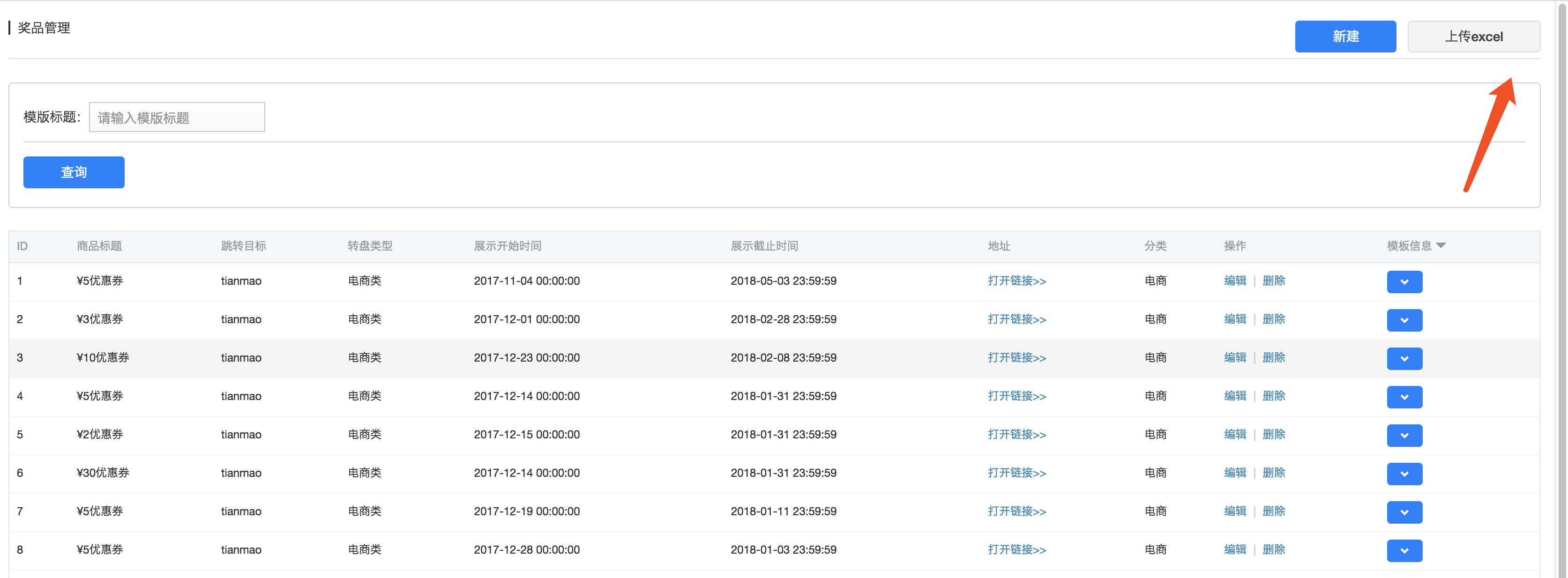
Excel上传文件并导入数据

# 1、前台显示



# 2、controller

## 解释：ArrayList<String> 表示每行的数据集合 List表示将那么多行集合到

@RequestMapping("uploadExcel")  
@ResponseBody  
public ResponseBean uploadExcel(MultipartFile file){  
 try {  
 POIExcelHelper excel=new POIExcelHelper();  
 List<ArrayList<String>> data= excel.read(file.getInputStream(),false);  
  
 gameService.excelImport(data);  
  
 } catch (Exception e) {  
 logger.error(e.getMessage(),e);  
 return ResponseBean.buildFailure("导入失败");  
 }  
  
 return ResponseBean.buildSuccess();  
}

# 3、service开始读取数据

## 1、读取数据并导入

@Override  
public void excelImport(List<ArrayList<String>> data) {  
 List<GamePrize> list = new ArrayList<>();  
 for (int i = 1; i < data.size(); i ++) {  
 GamePrize gamePrize = new GamePrize();  
 List<String> dd = data.get(i); 读取excel每行的数据  
 if(StringUtils.isEmpty(dd.get(0))||StringUtils.isEmpty(dd.get(1))||StringUtils.isEmpty(dd.get(2))||StringUtils.isEmpty(dd.get(3))||  
 StringUtils.isEmpty(dd.get(4))||StringUtils.isEmpty(dd.get(5))||StringUtils.isEmpty(dd.get(6))||StringUtils.isEmpty(dd.get(7))){  
 logger.info(i+"行标题是:"+dd.get(0));  
 continue;  
 }  
 DecimalFormat df = new DecimalFormat("#.0");  
 BigDecimal b1 = new BigDecimal(dd.get(2));  
 BigDecimal b2 = new BigDecimal(dd.get(4));  
 int discount = b1.subtract(b2).intValue();  
 gamePrize.setLink(dd.get(7));  
 gamePrize.setTitle("¥"+discount+"优惠券");  
 gamePrize.setTitle1("¥"+discount+"优惠券");  
 gamePrize.setIcon3(dd.get(1)+"\_300x300q90.jpg");  
 gamePrize.setTitle3(dd.get(0));  
 gamePrize.setTitle4(""); //成交量为""  
 gamePrize.setTitle5(df.format(b1));  
 gamePrize.setTitle6(df.format(b2));  
 gamePrize.setTitle2(dd.get(5).replace("-",".") + "-" + dd.get(6).replace("-","."));  
 gamePrize.setTarget(GamePrize.***TARGET\_TIANMAO***);  
 gamePrize.setTags1("1");  
 gamePrize.setIconType(GameUserPrize.***ICON\_4***);  
 gamePrize.setTemplateType(1);  
 gamePrize.setStatus(GamePrize.***STATUS\_ENABLE***);  
 gamePrize.setCdate(new Date());  
 gamePrize.setButtonTitle("立即领取");  
 gamePrize.setIcon1(oss\_host+"201710/7c4364a95c8c4050a8c5d4381cb23fe7.jpg"); //天猫图标OSS存储路径  
 gamePrize.setStartTime(DateHelper.getDateFirstTime(DateHelper.convertString2Date(dd.get(3),DateHelper.***YYYY\_MM\_DD***)).getTime());  
 gamePrize.setEndTime(DateHelper.getDateLastTime(DateHelper.convertString2Date(dd.get(6),DateHelper.***YYYY\_MM\_DD***)).getTime());  
 gamePrize.setShowEndTime(DateHelper.getDateLastTime(DateHelper.convertString2Date(dd.get(6),DateHelper.***YYYY\_MM\_DD***)).getTime());  
 gamePrize.setShowStartTime(DateHelper.getDateFirstTime(DateHelper.convertString2Date(dd.get(3),DateHelper.***YYYY\_MM\_DD***)).getTime());  
  
  
 list.add(gamePrize);  
 }  
 List<GamePrize> gamePrizeList = (List<GamePrize> )gamePrizeRepository.save(list);  
  
 //设置分类  
 List<GamePrizeCategory> gamePrizeCategoryList = new ArrayList<>();  
 for (GamePrize gamePrize : gamePrizeList) {  
 GamePrizeCategory prizeCategory = new GamePrizeCategory();  
 prizeCategory.setPrizeId(gamePrize.getPrizeId());  
 prizeCategory.setCategoryId(6l);  
 gamePrizeCategoryList.add(prizeCategory);  
 }  
 gamePrizeCategoryRepository.save(gamePrizeCategoryList);  
  
}

# 4、工具类

package com.duodian.admore.admin.utils;  
  
import org.apache.poi.hssf.usermodel.HSSFDateUtil;  
import org.apache.poi.hssf.usermodel.HSSFWorkbook;  
import org.apache.poi.ss.usermodel.Cell;  
import org.apache.poi.ss.usermodel.Row;  
import org.apache.poi.ss.usermodel.Sheet;  
import org.apache.poi.ss.usermodel.Workbook;  
import org.apache.poi.xssf.usermodel.XSSFWorkbook;  
  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.IOException;  
import java.io.InputStream;  
import java.text.DecimalFormat;  
import java.util.ArrayList;  
import java.util.List;  
  
/\*\*  
 \* 类名称：POIExcelUtil  
 \* 类描述：  
 \* 创建人：shiwei.li  
 \* 修改人：  
 \* 修改时间：2015/7/21 16:51  
 \* 修改备注：  
 \*  
 \* @version 1.0.0  
 \*/  
public class POIExcelHelper {  
  
 private int totalRows = 0;// 总行数  
 private int totalCells = 0;// 总列数  
  
 public POIExcelHelper() {  
 }  
  
 // 根据文件名读取excel文件  
 public List<ArrayList<String>> read(String fileName) {  
 List<ArrayList<String>> dataLst = new ArrayList<ArrayList<String>>();  
  
 // 检查文件名是否为空或者是否是Excel格式的文件  
 if (fileName == null || !fileName.matches("^.+\\.(?i)((xls)|(xlsx))$")) {  
 return dataLst;  
 }  
  
 boolean isExcel2003 = true;  
 // 对文件的合法性进行验  
 if (fileName.matches("^.+\\.(?i)(xlsx)$")) {  
 isExcel2003 = false;  
 }  
  
 // 检查文件是否存在  
 File file = new File(fileName);  
 if (file == null || !file.exists()) {  
 return dataLst;  
 }  
  
 try {  
 // 读取excel  
 dataLst = read(new FileInputStream(file), isExcel2003);  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 return dataLst;  
 }  
  
 // 根据流读取Excel文件  
 public List<ArrayList<String>> read(InputStream inputStream, boolean isExcel2003) {  
 List<ArrayList<String>> dataLst = null;  
 try {  
 // 根据版本选择创建Workbook的方式  
 Workbook wb = isExcel2003 ? new HSSFWorkbook(inputStream) : new XSSFWorkbook(inputStream);  
 dataLst = read(wb);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 return dataLst;  
 }  
  
 // 得到总行数  
 public int getTotalRows() {  
 return totalRows;  
 }  
  
 // 得到总列数  
 public int getTotalCells() {  
 return totalCells;  
 }  
  
 // 读取数据  
 private List<ArrayList<String>> read(Workbook wb) {  
 List<ArrayList<String>> dataLst = new ArrayList<ArrayList<String>>();  
  
 Sheet sheet = wb.getSheetAt(0);// 得到第一个shell  
 this.totalRows = sheet.getPhysicalNumberOfRows();  
 if (this.totalRows >= 1 && sheet.getRow(0) != null) {  
 this.totalCells = sheet.getRow(0).getPhysicalNumberOfCells();  
 }  
  
 // 循环Excel的行  
 for (int r = 0; r < this.totalRows; r++) {  
 Row row = sheet.getRow(r);  
 if (row == null) {  
 continue;  
 }  
  
 ArrayList<String> rowLst = new ArrayList<String>();  
  
 // 循环Excel的列  
 for (short c = 0; c < this.getTotalCells(); c++) {  
 Cell cell = row.getCell(c);  
 String cellValue = "";  
 if (cell == null) {  
 rowLst.add(cellValue);  
 continue;  
 }  
  
 // 处理数字型的,自动去零  
 if (Cell.***CELL\_TYPE\_NUMERIC*** == cell.getCellType()) {  
  
 // 在excel里,日期也是数字,在此要进行判断  
 if (HSSFDateUtil.isCellDateFormatted(cell)) {  
 cellValue = DateHelper.convertDate2String(cell.getDateCellValue());  
 } else {  
 cellValue = getRightStr(cell.getNumericCellValue() + "");  
 }  
 } else if (Cell.***CELL\_TYPE\_STRING*** == cell.getCellType()) {// 处理字符串型  
 cellValue = cell.getStringCellValue();  
 } else if (Cell.***CELL\_TYPE\_BOOLEAN*** == cell.getCellType()) {// 处理布尔型  
 cellValue = cell.getBooleanCellValue() + "";  
 } else {// 其它数据类型  
 cellValue = cell.toString() + "";  
 }  
  
 rowLst.add(cellValue);  
 }  
 dataLst.add(rowLst);  
 }  
 return dataLst;  
 }  
  
 // 正确地处理整数后自动加零的情况  
 private String getRightStr(String sNum) {  
 DecimalFormat decimalFormat = new DecimalFormat("#.000000");  
 String resultStr = decimalFormat.format(new Double(sNum));  
 if (resultStr.matches("^[-+]?\\d+\\.[0]+$")) {  
 resultStr = resultStr.substring(0, resultStr.indexOf("."));  
 }  
 return resultStr;  
 }  
  
// public static void main(String[] args) throws Exception {  
//  
// String fileName = "E:\\ExcelDemo\\电子资产使用情况.xls";  
// POIExcelHelper excelUtil = new POIExcelHelper();  
// List<ArrayList<String>> dataLst = excelUtil.read(fileName);// 读取excle，并把数据打包成list  
//  
// System.out.println(" --------------------------excel内容如下--------------------------------");  
//  
// for (ArrayList<String> innerLst : dataLst) {  
// StringBuffer rowData = new StringBuffer();  
// for (String dataStr : innerLst) {  
// rowData.append(",").append(dataStr);  
// }  
// if (rowData.length() > 0) {  
// System.out.println(" " + rowData.deleteCharAt(0).toString());  
// }  
// }  
//  
// System.out.println(" 总行数：" + excelUtil.getTotalRows() + " , 总列数：" + excelUtil.getTotalCells());  
//  
// }  
}